NASA CR-66681

EXAMINATION OF ACCELERATED FIRST-ORDER METHODS FOR AIRCRAFT FLIGHT PATH OPTIMIZATION

Henry J. Kelley
Walter F. Denham

Report No. 68-19
Contract NAS 1-7987
October 1968

ANALYTICAL MECHANICS ASSOCIATES, INC.
57 OLD COUNTRY ROAD
WESTBURY, N. Y. 11590

NASA CR-66681

EXAMINATION OF ACCELERATED FIRST ORDER METHODS FOR AIRCRAFT FLIGHT PATH OPTIMIZATION

Henry J. Kelley
Walter F. Denham

Report No. 68-19
Contract NAS 1-7987
October 1968

ANALYTICAL MECHANICS ASSOCIATES, INC.
57 OLD COUNTRY ROAD
WESTBURY, N. Y. 11590

SUMMARY

The reasons that accelerated first order optimization methods appear attractive for atmospheric trajectory work are reviewed and the possibilities for extending such methods to handle constraints are examined. Difficulties arising with the gradient projection technique, in conjunction with these methods, recommend the adoption of penalty function approximation for treatment of constraints. Parameterization via spline polynomials is investigated briefly, and is found promising when used in conjunction with a time scale stretching device.

Two sets of recommendations are offered for modification of a gradient projection supersonic transport trajectory optimization computer program, one a minimum revision featuring the continuous conjugate gradient algorithm, and the second a rather major revision incorporating splines, the Davidon algorithm and a difference adjoint scheme for circumventing truncation error effects. Listings summarizing program changes appropriate to these recommendations are then presented.

TABLE OF CONTENTS

INTRODUCTION

Numerical trajectory optimization for aircraft flight presents particular difficulty because the problem entails relatively complex equations, generally long flight times, and tendencies toward numerical error magnification. The classical indirect method, numerical integration of the coupled Euler/state system of differential equations, suffers from instability and severe error magnification, while first order direct methods exhibit slow convergence for many types of problem, especially those whose solutions are long trajectories. Second order direct methods become unduly complex in the case of atmospheric flight, and the linearized state/Euler system common to all of these is subject, more or less, to the same basic instability phenomenon as the indirect method.

For these reasons it is of interest to examine the possibilities offered by accelerated first order methods, although these are sufficiently new in trajectory optimization applications as not yet to have been proved out even in relatively easy types of problem. The methods referred to consist of the conjugate gradient method and the Davidon variable metric method, both of these being candidates if the variational problem is first made into a parameter optimization problem by one means or another, while only the first is a serious contender for treatment of the continuous case.

In the following, we first review the algorithms and then examine the possibilities for extension to constrained problems via the two standard techniques of projection and penalty function approximation. After this, the problem of parameterization is taken up with special attention to possible use of spline approximation. Finally, some recommendations are advanced on features to be incorporated into new computer programs and possible revisions and extensions of existing programs, specifically an existing gradient projection program for supersonic transport flight path optimization. These recommendations lean heavily upon recent research results of the authors and are somewhat tentative because of limited computational experience.

REVIEW OF THE ALGORITHMS

To find the minimum of a function f of an n-vector x, which is smooth enough to possess continuous second partial derivatives, a series of steps in so-called conjugate directions may be taken, meaning steps Δx_i orthogonal to gradient increments Δf_{x_i}

$$\Delta f_{x_j}^T \Delta x_i = 0 \qquad , \qquad i > j$$

If the magnitude of each step is taken such as to minimize $f(x)$ in the direction taken, the process converges to the minimum in n steps for quadratic f.

Conjugate directions are generated by the conjugate gradient algorithm

$$\Delta x_i = -\alpha\, p_i \qquad , \qquad p_1 = f_{x_1}$$

$$p_i = f_{x_i} + \frac{|f_{x_i}|^2}{|f_{x_{i-1}}|^2}\, p_{i-1} \qquad , \qquad i \geq 2$$

in which the scalar parameter α governs the magnitude of the step, and is obtained by seeking out the one-dimensional minimum of $f(x - \alpha p)$ regarded as a function of α.

The Davidon variable metric algorithm also generates conjugate directions, given by

$$\Delta x_i = -\alpha_i H_i f_{x_i}$$

$$H_i = H_{i-1} + \frac{\Delta x_{i-1} \Delta x_{i-1}^T}{\Delta x_{i-1}^T \Delta f_{x_{i-1}}} - \frac{H_{i-1} \Delta f_{x_{i-1}} \Delta f_{x_{i-1}}^T H_{i-1}}{\Delta f_{x_{i-1}}^T H_{i-1} \Delta f_{x_{i-1}}} \quad , \quad i \geq 2$$

Here H is a matrix and H_1 is specified as a positive definite symmetric matrix. The scalar α_i is again obtained by one-dimensional minimization.

These methods are described in Refs. 1 through 4 and the variational equivalent of the conjugate gradient method in Ref. 5.

There have been various methods proposed which attempt to approximate the second partial matrix or its inverse and to employ it in a Newton's method scheme, avoiding altogether ideas of conjugacy and the need for one-dimensional minimizations. Such schemes encounter difficulty in retaining positive definiteness of the metric and in magnification of numerical errors, consequently are not considered sufficiently well developed for serious consideration at present.

TREATMENT OF CONSTRAINTS

If the minimization problem includes subsidiary conditions of the form $g(x) = 0$, where g is an m-vector, neither method applies directly. One approach to handling of constraints is penalty function approximation, the formation and minimization of the function

$$\bar{f} = f + \frac{1}{2} \sum_{j=1}^{m} k_j g_j^2$$

where the constants $k > 0$ are taken "large." The solution of this problem approaches that of the original as $k_j \to \infty$ if both exist and are uniquely defined. This device effectively converts the problem to an unconstrained one, to which any of the conjugate direction algorithms is then applicable.

A second approach regarded as standard is gradient projection. In the case of functions $g(x)$ which are linear in x, the constraint surfaces are hyperplanes and their intersection a linear space of smaller dimension, $n - m$, and the scheme proceeds by using any of the gradient algorithms on the projection of the gradient vector upon this subspace. This procedure is clearly equivalent to elimination of variables by use of the constraint equations.

In the case of nonlinear constraints, projection employs the locally tangent constraint hyperplanes. If large steps are taken in the negative projected gradient direction, provision must be made to deal with large constraint "violations." This difficulty is ordinarily overcome with conventional gradient methods simply by limiting the step magnitude to that for which constraint violations remain within prescribed bounds, and later correcting the violations by some separate means. The same situation, met with conjugate direction methods which are inherently large step methods, poses greater difficulty and will next be examined.

In addition to the difficulty of terminating one-dimensional searches and correcting constraint violations, there is a more basic difficulty arising with projection when the use of conjugate direction methods is attempted. This is the preservation of conjugacy of steps in any sense, in view of the shifting from one step to the next of the subspace defined by the intersection of constraint tangent hyperplanes. Both questions are in the research category and outside the scope of the presently reported study. While the writers can offer a suggestion for a modification of the search to take constraint violations into account, they regard the conjugacy question as decidedly nontrivial, and note that successful treatment of <u>both</u> questions would be required for attainment of anything like n-step convergence.

Briefly, the search termination difficulty is that a one-dimensional minimum will generally not exist in a straight-line search violating the constraints unless the function f possesses a minimum in the absence of constraints, which is, of course, a highly undesirable restriction. Failure of the one-dimensional search to terminate on a minimum could be remedied in the early iterations by terminating on constraint violations reaching prescribed bounds, but eventually there must be a one-dimensional minimum attained on each successive step to support any sort of n-step convergence argument based upon conjugacy.

It is suggested that this difficulty might be circumvented by searching on the corrected function

$$f + \lambda g$$

where

$$\lambda \equiv - \frac{f_x A^T g_x}{g_x^T A g_x} = - \frac{df}{dg}$$

which contains a term correcting approximately for violations of a single constraint $g = 0$ based upon a linear trade-off defined by directional derivatives of the functions f and g along the normal to the constraint. The scalar λ becomes independent of the metric A as the constrained minimum is approached. It can be shown that if the constrained minimum is a "well defined" one, in the sense that the strengthened inequality

$$\delta x^T (f_{xx} + \lambda g_{xx}) \delta x > 0$$

for all δx such that

$$g_x^T \delta x = 0$$

applies, then the function $f + \lambda g$ will exhibit a minimum along the direction of search once a sufficiently small neighborhood of the constrained minimum has been reached. The correction term λg is a kind of linear penalty term that provides a good approximation to the "cost" of correcting a constraint violation. While a linear penalty scheme will generally be unsuccessful, the assumed closeness of the minimum and the search starting in the tangent hyperplane permit its use here.

SPLINE PARAMETERIZATION OF CONTINUOUS CONTROL PROGRAMS

The control variables in the usual aircraft flight equations are continuous functions of time. Most of the methods being discussed here require that these control functions be parameterized. There are two alternatives; one to superimpose a finite number of continuous functions defined over the whole interval, another to use different continuous functions in a sequence of subintervals. In the first case, one might use N orthogonal functions. In the second case, one would choose N subintervals and impose conditions at the junction times.

The great disadvantage of using complete interval functions is that it could easily take quite a large number of functions to adequately represent a particular short interval fluctuation. In the aircraft trajectory problem, the climb is both most critical and most sensitive, and some controls will have important rapid variations. The cruise interval, however, is flown nearly in equilibrium. The overall control histories will be far from any representation by a small number of simple orthogonal functions.

The use of N subintervals can also require large N if the choice of intervals and functions is crude. For example, using piecewise constant control in each computing interval would seem unwise, for with the same number of parameters, one could use continuous piecewise linear control. Still, the number of integration intervals tends to be significantly larger than parameter optimization programs can readily handle. It is important to keep the number of subintervals down to the order of ten or twenty, and it would seem that piecewise linear controls over these larger intervals would be too crude.

An attractive compromise between number of parameters and overall smoothness of the resulting control is the use of splines. For the aircraft trajectory problem, the cubic spline is especially appealing. In each subinterval, the function is a cubic polynomial. The coefficients are restricted so as to force the function and its first and second derivatives to be continuous at the junction points. With N subintervals, there are $N + 3$ parameters to be specified: $N+1$ values of the function itself at the junction (and end) points, plus two end conditions. These values plus the junction continuity conditions determine all the polynomial coefficients. Higher order splines with correspondingly greater smoothness at the junction points are also straightforward to calculate, but continuity of second derivatives would appear to be adequate for any of the aircraft control histories. The great advantage of the spline is that it allows significant short interval fluctuations without unduly increasing the total number of parameters required to represent the function.

Let us consider how a particular control function, say $u(t)$, is approximated by a cubic spline. Times $t_o, t_1, ---, t_N$ are chosen, then values $u_i = u(t_i)$. The spline approximation is described by

$$u(t) = M_{j-1}\frac{(t_j-t)^3}{6h_j} + M_j\frac{(t-t_{j-1})^3}{6h_j} + \left(u_{j-1} - \frac{M_{j-1}h_j^2}{6}\right)\frac{t_j-t}{h_j}$$

$$+ \left(u_j - \frac{M_j h_j^2}{6}\right)\frac{t-t_{j-1}}{h_j} \qquad , \qquad t_{j-1} \le t \le t_j$$

The M_j values may be seen to be second derivatives of u at t_j. The continuity conditions require them to satisfy

$$
\begin{bmatrix}
2 & \lambda_o & 0 & - & - & - & - & 0 & 0 \\
\mu_1 & 2 & \lambda_1 & - & - & - & - & 0 & 0 \\
0 & \mu_2 & 2 & - & - & - & - & 0 & 0 \\
 & & & & & & & & \\
 & & & & & & & & \\
0 & 0 & - & - & - & - & 2 & \lambda_{N-2} & 0 \\
0 & 0 & - & - & - & - & \mu_{N-1} & 2 & \lambda_{N-1} \\
0 & 0 & - & - & - & - & 0 & \mu_N & 2
\end{bmatrix}
\begin{bmatrix}
M_o \\ M_1 \\ M_2 \\ \\ \\ M_{N-2} \\ M_{N-1} \\ M_N
\end{bmatrix}
=
\begin{bmatrix}
d_o \\ d_1 \\ d_2 \\ \\ \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

where

$$
d_j = \frac{6}{h_j + h_{j+1}} \left[\frac{u_{j+1} - u_j}{h_{j+1}} - \frac{u_j - u_{j-1}}{h_j} \right] \quad , \quad j = 1, 2, ---, N
$$

and

$$
h_j = t_j - t_{j-1}
$$

$$
\lambda_j = \frac{h_{j+1}}{h_j + h_{j+1}} \quad , \quad \mu_j = 1 - \lambda_j \quad , \quad j = 1, 2, ---, N
$$

The end conditions enter through the first and last of the N+1 simultaneous equations.

In addition to the N+1 u_i values, two other numbers must be specified to allow solution of the M_j equations. The choice is non-unique. A simple procedure is to (arbitrarily) set λ_o and μ_N each equal to $\frac{1}{2}$, and let d_o and d_N be the two additional parameters. Writing the simultaneous equations in matrix form as

$$
A\,M = d
$$

the $(N+1) \times (N+1)$ matrix A is then completely known, and the $(N+1)$-vector d is determined by the u_i, d_o and d_N.

In general, the terminal time is not specified. One could treat this additional parameter directly, but it seems wiser to use the stretching device due to Long [7]. The complication of letting t_N vary directly is that it might become large, with u still forced to be cubic in the last interval $t_N - t_{N-1}$, or, perhaps worse, t_N might tend to become less than t_{N-1}. In either case, it would be necessary to change the number of parameters. In an ordinary gradient method, this would be easily done. In a conjugate gradient method, however, it would require restarting the algorithm and losing the beneficial conjugacy of the previous steps.

Long's suggestion was to normalize the time so as to keep the final time equal to one. This is done by a simple scaling, say $t = \beta t'$, when $t'_N = 1$. All differential equations are written in terms of t', with the parameter β appearing explicitly, e.g.

$$\frac{dV}{dt'} = \frac{dV}{dt}\frac{dt}{dt'} = \frac{dV}{dt}\beta$$

The input time values t'_i range from zero to one, and are fixed throughout the optimization. This stretching device keeps the relative spacing of the t_j intact, and simultaneously stretches the cubic of each interval. This seems decidedly preferable to adjusting only the last interval. The control is calculated as a function of t', which is immediately converted to a function of t after the optimization is complete.

For any gradient method, the calculation of the gradient is of central interest. In this case we have as many as four control functions, with $N+3$

values of u_i, d_o, d_N for the parameterization of each. The scaling parameter β is a single additional parameter, resulting in performance index and constraints dependent on $4(N+3)+1$ quantities. The gradient we seek, then, has up to $4(N+3)+1$ components.

We focus again now on the "typical" control function $u(t)$, spline parameterized through u_i, d_o, d_N. The gradient components corresponding to these parameters are found using a straightforward variation of the usual adjoint equation procedure. Let us write the equations of motion in vector form

$$\dot{y} = f(y, u, t)$$

where y includes position, velocity, mass and u represents the control. The adjoint differential equations (in matrix form) are then

$$\dot{\lambda} = -\left(\frac{\partial f}{\partial y}\right)^T \lambda$$

With time non-dimensionalized, we have

$$\frac{dy}{dt'} = \frac{dy}{dt}\frac{dt}{dt'} = \beta f(y, u, t)$$

$$\frac{d\lambda}{dt'} = \frac{d\lambda}{dt}\frac{dt}{dt'} = \beta \left(\frac{\partial f}{\partial y}\right)^T \lambda$$

Adapting the usual procedure, small variations δy about a reference trajectory satisfy

$$\frac{d}{dt'}(\lambda^T \delta y) = \beta \lambda^T \frac{\partial f}{\partial u} \delta u(t') + \lambda^T f \delta\beta$$

The $\lambda(t')$ and $\frac{\partial f}{\partial u}(t')$ functions appear as usual, leaving us only with the detail of expressing δu in terms of parameter variations. Examining the equation for u (now as a function of t'), we see a dependence on M_j, M_{j-1}, u_j and u_{j-1} in the interval $t'_{j-1} \leq t' \leq t'_j$. Note that the t'_j and h_j (now taken as $t'_j - t'_{j-1}$) are all fixed. In this interval, then,

$$\delta u = \frac{t'_j - t'}{6h_j}\left[(t'_j - t')^2 - h_j^2\right]\delta M_{j-1} + \frac{t' - t'_{j-1}}{6h_j}\left[(t' - t'_{j-1})^2 - h_j^2\right]\delta M_j$$

$$+ \frac{t'_j - t'}{h_j}\,\delta u_{j-1} + \frac{t' - t'_{j-1}}{h_j}\,\delta u_j$$

But recalling $A\,M = d$

$$M_i = \sum_{k=0}^{N} (A^{-1})_{ik}\, d_k \qquad\qquad i = 0, 1, ---, N$$

so that

$$\delta M_i = \sum_{k=0}^{N} (A^{-1})_{ik}\, \delta d_k$$

Further, δd_k, for $1 \leq k \leq N-1$, depends on δu_i values as follows:

$$\delta d_k = \frac{6}{h_k + h_{k+1}}\left[\frac{\delta u_{k+1} - \delta u_k}{h_{k+1}} - \frac{\delta u_k - \delta u_{k-1}}{h_k}\right]$$

By successively substituting for δM in terms of δd and for (most of) δd in terms of δu_i, we obtain an expression for δu entirely in terms of δu_i, δd_o and δd_N. In the interval $t'_{j-1} \leq t' \leq t'_j$

$$\delta u = \frac{t_j' - t'}{6h_j}\left[(t_j' - t')^2 - h_j^2\right]\left[\sum_{k=1}^{N-1} (A^{-1})_{j-1,k}\left(\frac{6}{h_k + h_{k+1}}\right)\left(\frac{\delta u_{k+1} - \delta u_k}{h_{k+1}} - \frac{\delta u_k - \delta u_{k-1}}{h_k}\right)\right.$$

$$\left. + (A^{-1})_{j-1,0}\,\delta d_o + (A^{-1})_{j-1,N}\,\delta d_N\right]$$

$$+ \frac{t' - t_{j-1}'}{6h_j}\left[(t' - t_{j-1}')^2 - h_j^2\right]\left[\sum_{k=1}^{N-1} (A^{-1})_{j,k}\left(\frac{6}{h_k+h_{k+1}}\right)\left(\frac{\delta u_{k+1} - \delta u_k}{h_{k+1}} - \frac{\delta u_k - \delta u_{k-1}}{h_k}\right)\right.$$

$$\left. + (A^{-1})_{j,0}\,\delta d_o + (A^{-1})_{j,N}\,\delta d_N\right] + \frac{t_j' - t'}{h_j}\,\delta u_{j-1} + \frac{t' - t_{j-1}'}{h_j}\,\delta u_j$$

From this expression for δu, the coefficients of each δu_i and of δd_o and δd_N are selected by inspection and may be denoted as

$$\frac{\partial u}{\partial u_i}\,,\ \frac{\partial u}{\partial d_o}\,,\ \frac{\partial u}{\partial d_N}$$

respectively. Then

$$\frac{d}{dt'}(\lambda^T \delta y) = \beta\lambda^T \frac{\partial f}{\partial u}\left[\sum_{i=0}^{N} \frac{\partial u}{\partial u_i}\delta u_i + \frac{\partial u}{\partial d_o}\delta d_o + \frac{\partial u}{\partial d_N}\delta d_N\right] + \lambda^T f\,\delta\beta$$

Upon integrating, choosing $\lambda^T(t_N') = \frac{\partial Q}{\partial y(t_N)}$, and noting that $y(t_o)$ is assumed fixed, we have

$$\delta Q = \sum_{i=0}^{N}\left[\int_0^1 \beta\lambda^T \frac{\partial f}{\partial u}\frac{\partial u}{\partial u_i}\,dt'\right]\delta u_i + \left[\int_0^1 \beta\lambda^T \frac{\partial f}{\partial u}\frac{\partial u_i}{\partial d_o}\,dt'\right]\delta d_o$$

$$+ \left[\int_0^1 \beta\lambda^T \frac{\partial f}{\partial u}\frac{\partial u_i}{\partial d_N}\,dt'\right]\delta d_N + \left[\int_0^1 \lambda^T f\,dt'\right]\delta\beta$$

Now, by inspection, we have $\frac{\partial Q}{\partial u_i}$, $\frac{\partial Q}{\partial d_o}$, $\frac{\partial Q}{\partial d_N}$ and $\frac{\partial Q}{\partial \beta}$ as the coefficients of δu_i, δd_o, δd_N and $\delta\beta$ respectively.

Programming alterations required by the spline approximation are not especially complex, being mostly add-on in nature. The equations of motion and the adjoint equations each are multiplied by the constant β due to time non-dimensionalization. In addition to the $\frac{\partial f}{\partial u}$ vectors already programmed, the $\frac{\partial u_i}{\partial u}$, etc. expressions must be programmed. The integrals for $\frac{\partial Q}{\partial u_i}$, etc. must also be programmed.

If all constraints are treated by penalty functions, Q is the augmented performance index. When additional functions are considered, as in gradient projection, the same equations are used. Only the $\lambda(t')$ solutions differ, requiring additional evaluations of the same integrals.

Bounds on a control function may be included with spline approximations, either with integral penalty functions or directly, leading to side constraints on the u_i, d_o, d_N values. The integral penalty function needs no further elaboration. Consider next a direct constraint treatment.

Suppose that $u(t')$ is to be bounded by u_{max} and u_{min}. (These bounds could be different in each interval, but the situation would be substantially complicated if the bounds were to vary arbitrarily in time.) For the cubic employed for the interval $t'_{j-1} \leq t' \leq t'_j$, u has a maximum value. This value can be determined by setting $\frac{du}{dt'}$ equal to zero and solving for t'. As u is cubic, $\frac{du}{dt'}$ will be quadratic in t'; hence there will be possibly two zeros of $\frac{du}{dt'}$. If both roots are real, one corresponds to a local minimum and one to a local maximum. (The formulas for these times may be programmed explicitly.) Either the maximum, if it occurs within the interval, or u_{j-1} or u_j is the largest u in the interval. These three values must be compared against u_{max}, the analogous three against u_{min}. If, for any intended forward integration, this largest value is greater than u_{max} (at time, say, t^*), some adjustment must be made in u_i, d_o, d_N. As may be readily observed, $u(t)$ is linear in u_i and M_i; hence in u_i, d_o, d_N. Thus, each $u(t^*)$ value which would have exceeded u_{max} results in a linear constraint involving u_i, d_o, d_N. The exact analogy holds for u_{min}. What could happen, then, is a collection of several such linear constraints coming from different intervals. The parameterized optimization problem, then, is one with a number of linear inequality constraints. As is usual in such problems, the question of which constraints are to be equalities and which ignored must be considered at each cycle. For this reason it may be preferable to use integral penalty functions, which appears to be necessary (or at least desirable) anyway, if a conjugate gradient method is to be used.

In the interest of numerically verifying the workability of cubic spline parameterization, a complete function minimization program has been prepared. The dynamic system is academic in nature. A linear system with quadratic performance index was chosen, so that it would be possible to compare the optimized spline against the exact optimum (obtainable by a direct computer calculation) if desired.

The system is described by

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = u$$

$$\text{Minimize } \left\{[y_1(t_N) - 1]^2 + [y_2(t_N)]^2 + \int_0^T [y_1^2 + (\cos^2 6t)\, y_2^2 + (\sin^2 6t)\, u^2]\, dt\right\}$$

The problem is to choose $u(t)$ and the final time T to minimize the performance index indicated. The initial y_1 would be chosen large enough to insure $T > 0$.

No change in the parameter optimization algorithm was required to use spline approximation of the control. Hence, the new programming required involved mainly the evaluation of the performance function gradient with respect to the spline parameters (and T). The program has been coded and is in the initial checkout phase as of this writing.

RECOMMENDED FEATURES FOR AIRCRAFT FLIGHT PATH OPTIMIZATION COMPUTER PROGRAMS

We recommend two combinations of features for possible incorporation into existing first order computer programs. The first of these is selected to provide more rapid convergence with the least revision of existing programs. The second describes our choice as to the most promising program for accelerating convergence, in terms both of sureness and efficiency.

In the first combination, the following are unchanged:

- The equations of motion
- The adjoint equations, and the impulse response functions
- The tabular representation of continuous control variables
- The numerical integration schemes for both forward and backward integrations

The essential change, use of the (continuous conjugate gradient algorithm, requires

- Penalty function treatment of all constraints
- Storage of an additional vector of time functions, dimensionally equal to the (continuous) control vector
- Extended precision arithmetic (16 or more decimal digits)

The second recommendation suggests combining the most promising new developments in function minimization techniques into an essentially new program. Some features are moderately well established, others are very promising on theoretical grounds, although not well tested numerically. It appears that some separate pretesting of the newest developments would be preferable to an immediate move to a complete new program. This second combination modifies nearly every portion of a conventional gradient program. The two main innovations are the

> Parameterization of the control functions to allow use of the Davidon function minimization algorithm
>
> Use of a compatible adjoint scheme to achieve much greater numerical accuracy of the payoff function gradient (Ref. 8)

The former, in its recommended form, requires

> The use of a (cubic) spline approximation routine to allow control variable evaluations in terms of N+3 parameters (each).
>
> The Davidon variable metric algorithm routine (strictly algebraic)
>
> The use of Long's stretch factor device to nondimensionalize the time and conveniently allow unspecified terminal time
>
> Penalty function treatment of constraints, with refinement by the approximation to Newton's method of Ref. 9.

REVISIONS REQUIRED FOR ATMOSPHERIC FLIGHT GRADIENT PROJECTION COMPUTER PROGRAMS

In the following, we list the revisions required to implement the two sets of recommendations just outlined for existing gradient projection computer programs. The three-dimensional atmospheric flight program for supersonic transport trajectory optimization, described in Ref. 11, is of particular interest since it is in current use by NASA Langley Research Center, the sponsor of the present study, and consumes large amounts of computer running time in its present form.

Revisions for Combination One

1. Extend arithmetic precision to sixteen or more decimal digits.
2. Code penalty function transversality conditions for terminal adjoint variable determination.
3. Reduce the number of backward adjoint system evaluations to one, using these terminal conditions.
4. Delete the evaluation of projection integrals, terminal constraint multiplier evaluations, and projection step-size logic.
5. Provide additional storage for control functions corresponding to the second member of the conjugate gradient formula.
6. Program the conjugate gradient algorithm, including evaluation of auxiliary integrals needed and provision for restart.
7. Add a one-dimensional search routine (e.g., Ref. 12) and orthogonality correction (Ref. 10).
8. Program checkout and provision for diagnostic tests.

Revisions for Combination Two

1 & 2. Same as for preceding Combination One.

3. Provide for storage of the state vector as a function of time at fixed integration time steps.

4. Derive difference adjoint equations for the integration scheme used and code, providing for a single backward adjoint "integration."

5. Delete control function tables and replace with cubic spline representation.

6. Incorporate time scale stretch factor.

7. Evaluate gradient with respect to spline parameters and code.

8. Program accelerated gradient algorithm, including a one-dimensional search, orthogonality correction, penalty coefficient adjustment, and refinement scheme, possibly adopting routines developed for the work of Refs. 8, 10 and 12.

9. Program checkout and provision for diagnostic tests.

RATIONALE AND OUTLOOK FOR THE RECOMMENDED COMBINATIONS OF FEATURES

Both recommended combinations include penalty functions and extended precision arithmetic, based upon the experience of Refs. 9 and 10. We have been somewhat reluctant to conclude that the gradient projection scheme does not extend to nonlinear constraints when accelerated gradient methods are used and, accordingly, have devoted some attention to review of the problems encountered and possible means of circumventing them, as reported in an earlier section. Whatever may be possible here is clearly a matter for research and experimentation on simple examples, and any scheme emerging will be well behind the penalty function version, which has been subjected to extensive testing in parameter optimization applications work.

For Combination One, the programming requirements are modest. Penalty function treatment of terminal as well as in-flight constraints is a trivial alteration. The additional function storage is simply in parallel with existing storage. Depending on the computer software, the extended precision could be as simple as a single card insertion, or as troublesome as redefining every variable as a double precision quantity. The clear advantage of Combination One is the retention of most of the existing programming with modest additions which, taken alone, should not be disruptive. The additional storage required, however, may cause overflow in computers having the standard 32K eight-digit word capacity.

The eventual gain using the conjugate gradient routine is uncertain. In general, it would be safe to say that the number of cycles required to reach a given value (close to the minimum) of the augmented performance index (constraints adjoined as penalty terms) will be less with a conjugate gradient than with a "regular" gradient program. Computer time comparisons cannot yet be safely made, primarily because the high accuracy required by the method may force reduction in computing intervals to bring truncation error to acceptable levels. While the continuous conjugate gradient method has been used to a limited extent in modest test problems with some success, there is no experience with programs approaching the complexity and extent of the SST program, and so there is an element of gamble even in proceeding with this relatively modest modification.

Each of the several requirements of Combination Two has its own neatly separate programming and no one item is particularly complex or formidable. As a result of the non-dimensionalization of time, the stretching parameter becomes a simple multiplicative factor in the equations of motion, and the straightforward means of making use of existing programming would be to evaluate the time derivatives as presently done, then multiply each by the common stretch factor.

The most substantial new programming required would be the difference adjoint routine which produces compatibility between the computed function samples and the computed gradient of the function, in the sense that both reflect the approximation made in integrating the state via whatever finite difference scheme has been adopted. The details have been worked out for the Kutta-Merson integrator in Ref. 8, and the corresponding derivation for the Runge-Kutta case should be quite similar. Restating the case for the difference, or "compatible," adjoint, it is that the partial derivatives of the terminal state with respect to trajectory parameters are accurate representations of the partials of the terminal state as calculated through the integration model.

As with Combination One, extended precision arithmetic is necessary to exploit the accelerated convergence potential, because of the vulnerability of conjugate direction algorithms to numerical error (Ref. 10). The compatible adjoint, however, promises to alleviate the need for smaller computing intervals, in fact to suggest a sequence of two or even three optimizations, starting with a very large computing interval and finishing with the normally acceptable interval. This is because the compatible adjoint routine completely compensates for truncation error in the trajectory, as its derivation explicitly involves the forward integration routine equations.

By using a finite number of parameters (order ten) for each control function, convergence to a minimum will eventually be quadratic, and it is also likely that the neighborhood of the minimum will be reached in fewer cycles than the conventional gradient or gradient projection algorithm. The compatible adjoint may, through use of larger computing intervals, reduce the total computing time by a factor of three or more. The parameterization features are by now sufficiently understood so that no major risk is likely in programming them. The power of the compatible adjoint routine is not well tested, however, and it is recommended that a program for a simple problem might well be first prepared for experimentation.

CONCLUDING REMARKS

The recommendations advanced herein reflect the current judgment of the writers, based upon their recent research. It is thought that trials on a somewhat smaller scale than the three-dimensional, multi-control, multi-constraint supersonic transport trajectory optimization problem might be appropriate before undertaking revisions of the SST program, in order to prove out the proposed combinations of features. This might apply particularly to the second combination, which incorporates a considerable number of innovations. Possibly a two-dimensional, atmospheric, single control variable (attitude) model would be of about the proper degree of complexity, yet result in a computer program having some usefulness in itself, apart from its value as a test bed.

REFERENCES

1. Beckman, F. S.; "The Solution of Linear Equations by the Conjugate Gradient Method," in Mathematical Methods for Digital Computers, A. Ralston and H. S. Wilf, Editors, Wiley, 1960.

2. Fletcher, R. and Reeves, C. M.; "Function Minimization by Conjugate Gradients," Computer Journal, July 1964.

3. Davidon, W. C.; "Variable Metric Method for Minimization," Argonne National Laboratory Report ANL-5990 Rev., November 1959.

4. Fletcher, R. and Powell, M. J. D.; "A Rapidly Convergent Descent Method for Minimization," Computer Journal, July 1963.

5. Lasdon, L. S., Mitter, S. K. and Waren, A. D.; "The Conjugate Gradient Algorithm for Optimal Control Problems," IEEE Transactions on Automatic Control, April 1967.

6. Ahlberg, J. H., Nilson, E. N. and Walsh, J. L.; The Theory of Splines and Their Applications, R. Bellman, Editor, Academic Press, 1967.

7. Long, R. S.; "Newton-Raphson Operator: Problems with Undetermined End Points," AIAA Journal, July 1965.

8. Kelley, H. J. and Denham, W. F.; "Modeling and Adjoints for Continuous Systems," Second International Conference on Computing Methods in Optimization Problems, San Remo, Italy, September 1968; to appear in the Journal of Optimization Theory and Applications, 1969.

9. Kelley, H.J., Denham, W.F., Johnson, I.L. and Wheatley, P.O.; "An Accelerated Gradient Method for Parameter Optimization with Nonlinear Constraints," Journal of the Astronautical Sciences, July-August 1966.

10. Kelley, H.J. and Myers, G.E.; "Conjugate Direction Methods for Parameter Optimization," 18th Congress of the International Astronautical Federation, Belgrade, Yugoslavia, September 1967.

11. Stein, L.H., Matthews, M.L. and Frenk, J.W.; "STOP – A Computer Program for Supersonic Transport Trajectory Optimization," NASA CR-793, 1967.

12. Johnson, I.L. and Myers, G.E.; "One-Dimensional Minimization Using Search by Golden Section and Cubic Fit Methods," NASA Manned Spacecraft Center Internal Note 67-FM-172, November 1967.